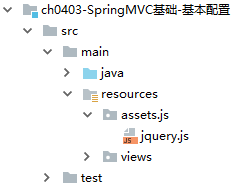
# Spring MVC基础-常用配置

Spring MVC 的定制配置需要我们的配置类继承一个WebMvcConfigurerAdapter 类，并在此类使用＠EnableWebMvc 注解，来开启对Spring MVC 的配置支持，这样我们就可以重写这个类的方法，完成我们的常用配置。

## 静态资源映射

程序的静态文件（ js 、css 、图片）等需要直接访问，可以在配置里重写addResourceHandlers 方法来实现。我们在src/main/resources 下建立assets/js 目录，并复制一个jquery.js到此目录下。



## 拦截器配置

拦截器（ Interceptor ）实现对每一个请求处理前后进行相关的业务处理，类似于Serviet 的Filter 。可让普通的Bean 实现Hanlderlnterceptor 接口或者继承HandlerinterceptorAdapter 类来实现自定义拦截器。：通过重写WebMvcConfigurerAdapter的addInterceptors 方法来注册自定义的栏截器。

## @ControllerAdvice

通过@ControllerAdvice ，我们可以将对于控制器的全局配置放置在同一个位置，注解了@Controller 的类的方法可使用@ExceptionHandler、@InitBinder、@ModelAttribute 注解到方法上，这对所有注解了＠RequestMapping 的控制器内的方法有效。

@ExceptionHandler：用于全局处理控制器里的异常。

@InitBinder：用来设置WebDataBindcr, WebDataBinder用来自动绑定前台请求参数到Model 中。

@ModelAttribute：@ModelAttribute 本来的作用是绑定键值对到Model里，此处是让全局的@RequestMapping都能获得在此处设置的键值对。

## 其他配置

### 快捷的ViewController

配置页面转向的时候使用的如下代码

|  |
| --- |
| @RequestMapping(“/index”)  public String hello() {  return “index”;  } |

此处无任何业务处理，只是简单的页面转向，写了至少三行有效代码：在实际开发中会涉及大量这样的页面转向，若都这样写会很麻烦，我们可以通过在配置中重写addViewControllers来简化配置，这样实现的代码更简沽，管理更集中。

|  |
| --- |
| public void addViewControllers(ViewControllerRegistry registry) {  registry.addViewController(“/index”).setViewName(“/index”);  } |

### 路径匹配参数配置

在Spring MVC 中，路径参数如果带“．”的话，“．”后面的值将被忽略，例如，访问http://localhost: 8080/springmvc/anno/pathvar/xx. yy ，此时“．”后面的yy 被忽略。通过重写configurePathMatch 方法可不忽略“．”后面的参数，这时再访问http://localhost: 8080/springmvc/anno/pathvar/xx. yy，就可以接受“．”后面的yy 了，代码如下。

|  |
| --- |
| public void configurePathMatch(PathMatchConfigurer configurer) {  configure.setUserSuffixPattenMatch(false);  } |

更多配置可以查看WebMvcConfigurerAdapter 类。